In the Claims:

1. (Original) A method of inserting buffers in a circuit design, comprising the steps of:

preparing a physical hierarchy of the circuit design with placed macros;

performing global routing on the physical hierarchy;

determining a number of buffers to be inserted on each edge of nets of the global routing for boosting timing performance of the nets;

calculating a position for each buffer; and

inserting a buffer configured to boost timing performance at each calculated position.

2. (Original) The method according to Claim 1, wherein said buffers are inverters.

3. (Original) The method according to Claim 1, wherein said buffers are repeaters.

4. (Currently amended) The method according to Claim 1, wherein said step of ~~calculating intervals~~ determining a number of buffers comprises the steps of:

identifying a set of at least one edge in said nets for inserting buffers; and

determining an optimal number of buffers to be inserted on each edge.

5. (Original) The method according to Claim 4, wherein:

said step of determining an optimal number of buffers comprises the step of,

calculating, for each edge, the optimal number of buffers based on an optimal timing for the edge, a delay of the edge, and an impedance of the edge.

6. (Currently amended) The method according to Claim 5, wherein:

said step of calculating the optimal number of buffers comprises calculating

$$C_x^{\,i} = T_{opt} - D^{i-1} - \frac{\left(R^i_{eq} + \frac{1}{2}R^i\right)C^i}{(R_{eq}^{\,i} + R^i)f^i}$$

;

where $C_x^{\,i}$ is a capacitance contribution of a branch of a merged segment on an edge as seen by a driving node of the ~~branch~~ segment;

T_{opt} is the delay of an optimal stage;

[[D]] D^i is delay of the edge;

$R_{eq}^{\,i}$ is an equivalent resistance of the merged segment ~~segments of a corresponding branch segment~~;

[[f]] f^i is fanout of ~~a corresponding branch segment~~ the branch;

[[R]] R^i is a resistance of the edge; and.

[[C]] C^i is a capacitance of the edge.

7. (Currently amended) The method according to Claim 5, wherein:

said step of calculating the optimal number of buffers includes the steps of:

determining a uniform load distribution for all ~~connected~~ branches connected to the edge; and

adjusting for a delay introduced by the inserted buffers.

8. (Currently amended) The method according to Claim 7, wherein:

said step of determining a uniform load distribution comprises calculating a stage delay for each branch, comprising,

$$D^i = R^i\left[\frac{C^i}{2} + (f_i - 1)C_x^{\,i}\right] + R^i{}_{eq}[C^i + (f_i - 1)C_x^{\,i}];$$

where:

D^i is ~~the delay of the stage~~ a stage delay for branch *i*;

R^i is resistance of ~~the~~ branch *i*;

[[R_{eq}^i]] R^i_{eq} is an equivalent resistance of merged segments of branch *i*;

[[Ci]] C^i is capacitance of ~~the~~ branch *i*; and

[[Fi]] f^i is a fanout of ~~the~~ branch *i*.

9. (Currently amended) The method according to Claim 1, further comprising the steps of:

uniformly distributing a capacitance of each branch of the nets at a corresponding branch point;

determining a load at each branch point; and

checking if a buffer inserted at each branch point is capable of handling the ~~loaded~~ load determined for that branch point.

10. (Original) The method according to Claim 9, wherein said steps of determining a number of buffers and calculating a position of each buffer comprises:

identifying a driver of a net to have buffers inserted;

performing a breadth first search (BFS) of the net starting at the identified net driver and processing all connected edges;

performing a depth first search (DFS) starting at a next stage in the net; and

for each edge, determining a number of buffers to be inserted on each edge of nets of the global routing for boosting timing performance of the nets, and calculating a position for each buffer.

11. (Original) The method according to Claim 10, further comprising the steps of:

determining an actual capacitance of wires added to a branch at a branch point in a net;

using the actual capacitance in determining the number and position of buffers; and

redistributing capacitance comprising a difference between a calculated branch capacitance and the actual capacitance to other segments at said branch point.

12. (Original) The method according to Claim 10, further comprising the steps of:

summing of resistances in all segments between a driver to a current segment being processed;

determining a sum of delays caused by the summed resistances; and

passing the summed resistances and delays on to a next segment to be processed.

13. (Original) The method according to Claim 1, wherein:

said method is embodied in a set of computer instructions stored on a computer readable media;

said computer instructions, when loaded into a computer, cause the computer to perform the steps of said method.

14. (Currently amended) The method according to Claim 13, wherein said computer ~~instruction~~ instructions are compiled computer instructions stored as an executable program on said computer readable media.

15. (Cancelled)

16. (Currently amended) The method according to Claim 1, further comprising the steps of:

identifying a set of staircase edges in the circuit design;

forming a merged segment of all segments in the staircase at a preselected layer;

scaling a length of each staircase segment by a ratio taking into account parameters of the staircase and the merged ~~layer~~ segment;

determining if any of the segments ~~can~~ will be sped up using an inserted buffer; and

inserting buffers on the merged ~~edge~~ segment if the merged segment ~~can be~~ is sped up by the insertion.

17. (Original) The method according to Claim 16, wherein said ratio comprises a ratio of a segment per unit length resistance and a per unit length resistance of the merged layer.

18. (Currently amended) The method according to Claim 16, wherein said step of determining if any of the segments ~~can~~ will be sped up comprises, determining if the ratio of a pure wire delay of the merged segment to that of an isolated buffer delay is greater than or less than 1.

19. (Currently amended) The method according to Claim 16, wherein said step of inserting buffers comprises inserting buffers at a distance of l_{crit} from one of a start of the merged segment and a ~~preceeding~~ preceding buffer.

20. (Withdrawn) A method of correcting polarity with a minimized number of inverters in at least one path within a network, comprising the steps of:

marking all branch nodes with a polarity of a signal emanating from a driver up to the branch node being marked;

marking all sinks with a polarity of a signal emanating from a driver up to the branch node being marked;

traversing the network from each sink to an immediate branch node;

calculating a cost of correcting polarity of each sink;

carrying backwards the calculated cost to each sink; and

repeating said steps of traversing, calculating, and carrying until a root of the network is reached; and

forward visiting the network and inserting inverters to fix the polarity.

21. (Withdrawn) The method according to Claim 20, wherein said step of calculating a cost comprises:

determining a minimum cost between each of,

fixing the polarity on the segment driving the branch node, and

fixing the polarity on branches stemming out of the branch node; and

if downstream nodes of the branch node have a correct polarity, then storing zero at the branch node and carried backwards to an upstream branch node.

22. (Withdrawn) The method according to Claim 20, further comprising the step of:

storing a directive to indicate whether the minimum cost is associated with inserting an inverter on a trunk feeding the branch point or whether the inverter(s) are inserted on downstream segment(s).

23. (Withdrawn) The method according to Claim 20, wherein said cost is 1 or 0 depending on whether polarity is even or odd.

24. (Withdrawn) The method according to Claim 20, wherein:

said method is embodied in a set of computer instructions stored on a computer readable media;

said computer instructions, when loaded into a computer, cause the computer to perform the steps of said method.

25. (Withdrawn) The method according to Claim 24, wherein said computer instruction are compiled computer instructions stored as an executable program on said computer readable media.

26. (Withdrawn) The method according to Claim 20, wherein said method is embodied in a set of computer readable instructions stored in an electronic signal.

27. (New) A method of inserting buffers in a circuit design, comprising the steps of:

preparing a physical hierarchy of the circuit design with placed macros;

performing global routing on the physical hierarchy;

determining a number of buffers to be inserted on each edge of nets of the global routing for boosting timing performance of the nets by calculating, for each edge, the optimal number of buffers based on an optimal timing for the edge, a delay of the edge, and an impedance of the edge;

identifying a set of at least one edge in said nets for inserting buffers;

determining an optimal number of buffers to be inserted on each identified edge; and

inserting the optimal number of buffers configured to boost timing performance at each of calculated positions;

wherein:

each edge comprises a plurality of branches; and

said step of calculating the optimal number of buffers comprises calculating a capacitance C_x for each branch, comprising,

$$C_x^{\,i} = T_{opt} - D^{i-1} - \frac{\left(R^i_{eq} + \frac{1}{2}R^i\right)C^i}{(R_{eq}^{\,i} + R^i)f^i}$$

;

where,

C_x^i is a capacitance contribution of a branch *i* as seen by a driving node of branch *i*,

T_{opt} is a delay of an optimal stage,

D^i is delay of the edge,

R_{eq}^i is an equivalent resistance of all merged segments corresponding to branch *i*,

f^i is fanout of a corresponding branch *i*,

R^i is a resistance of branch *i*, and

C^i is a capacitance of branch *i*.

28. (New) The method according to Claim 27, wherein:

said method is embodied in a set of computer instructions stored on a computer readable media;

said computer instructions, when loaded into a computer, cause the computer to perform the steps of said method.

29. (New) A method of inserting buffers in a circuit design, comprising the steps of:

preparing a physical hierarchy of the circuit design with placed macros;

performing global routing on the physical hierarchy;

determining a number of buffers to be inserted on each edge of nets of the global routing for boosting timing performance of the nets;

calculating a position for each buffer by identifying a set of at least one edge in said nets for inserting buffers and determining an optimal number of buffers to be inserted on each edge; and

inserting a buffer configured to boost timing performance at each calculated position;

wherein:

determining an optimal number of buffers comprises, calculating, for each edge, the optimal number of buffers based on an optimal timing for the edge, a delay of the edge, and an impedance of the edge;

determining an optimal number of buffers further comprises determining a uniform load distribution for all connected branches and adjusting for a delay introduced by the inserted buffers; and

said step of determining a uniform load distribution comprises calculating a stage delay for each branch, comprising,

$$D^i = R^i\left[\frac{C^i}{2} + (f_i - 1)C_x{}^i\right] + R^i{}_{eq}[C^i + (f_i - 1)C_x{}^i]$$

where,

D^i is a stage delay for branch *i*,

R^i is resistance of branch *i*,

$R^i{}_{eq}$ is an equivalent resistance of all merged segments of branch *i*,

C^i is capacitance of branch *i*, and

f^i is a fanout of branch *i*.

30. (New) The method according to Claim 29, wherein:

said method is embodied in a set of computer instructions stored on a computer readable media;

said computer instructions, when loaded into a computer, cause the computer to perform the steps of said method.

31. (New) A method of inserting buffers in a circuit design, comprising the steps of:

preparing a physical hierarchy of the circuit design with placed macros;

performing global routing on the physical hierarchy;

determining a number of buffers to be inserted on each edge of nets of the global routing for boosting timing performance of the nets;

calculating a position for each buffer;

inserting a buffer configured to boost timing performance at each calculated position;

uniformly distributing a capacitance of each branch of the nets at a corresponding branch point;

determining a load at each branch point; and

checking if a buffer inserted at each branch point is capable of handling the load determined for that branch point;

wherein said steps of determining a number of buffers and calculating a position of each buffer comprises,

identifying a driver of a net to have buffers inserted,

performing a breadth first search (BFS) of the net starting at the identified net driver and processing all connected edges,

performing a depth first search (DFS) starting at a next stage in the net,

for each edge, determining a number of buffers to be inserted on each edge of nets of the global routing, wherein the buffers are for boosting timing performance of the nets, and

calculating a position for each buffer.

32. (New) The method according to Claim 31, wherein the calculated position for each buffer position is based on a combination of stage delay, branch resistance, resistance of merged branch segments, branch capacitance, and branch fanout.

33. (New) The method according to Claim 31, wherein:

said method is embodied in a set of computer instructions stored on a computer readable media;

said computer instructions, when loaded into a computer, cause the computer to perform the steps of said method.

34. (New) The method according to Claim 31, further comprising the steps of:

determining an actual capacitance of branch added wires at a branch point in a net;

using the actual capacitance in determining the number and position of buffers; and

redistributing capacitance comprising a difference between a calculated branch capacitance and the actual capacitance to other segments at said branch point.

35. (New) The method according to Claim 31, further comprising the steps of:

summing of resistances in all segments between a driver to a current segment being processed;

determining a total delay caused by the summed resistances; and

passing the summed resistances and total delay on to a next segment to be processed.